

Reasoning about Body-Parts Relations for Sign Language Recognition.

Marc Martínez Camarena* · José Oramas M.* ·
Mario Montagud Climent · Tinne Tuytelaars

Received: date / Accepted: date

Abstract Over the years, hand gesture recognition has been mostly addressed considering hand trajectories in isolation. However, in most sign languages, hand gestures are defined on a particular context (body region). We propose a pipeline to perform sign language recognition which models hand movements in the context of other parts of the body captured in the 3D space using the MS Kinect sensor. In addition, we perform sign recognition based on the different hand postures that occur during a sign. Our experiments show that considering different body parts brings improved performance when compared to other methods which only consider global hand trajectories. Finally, we demonstrate that the combination of hand postures features with hand gestures features helps to improve the prediction of a given sign.

Keywords Hand gesture recognition · sign language recognition · relational learning · classification.

This work is partially supported by FWO project G.0.398.11.N.10 “Multi-camera human behavior monitoring and unusual event detection”, KU Leuven GOA project CAMETRON, the “Fondo Europeo de Desarrollo Regional” (FEDER) and the Spanish Ministry of Economy and Competitiveness, under its R&D&i Support Program in project with ref TEC2013-45492-R.

Marc Martínez Camarena, Mario Montagud Climent,
IMM, Universidad Politécnica de Valencia (UPV), Spain.

José Oramas M., Tinne Tuytelaars
KU Leuven, ESAT-PSI, iMinds Belgium.

* First two authors had an equal contribution to this work.

1 Introduction

Hearing-impaired people as a community consider themselves a minority who communicates differently rather than a group of disabled people. Unfortunately, in some countries, this minority faces difficulties during their teaching/learning process. One of the most critical factors is the low teacher-student ratio, which directly affects the learning of communication skills by young students. As a consequence this hampers the possibility of the student for self-learning. Hence there is a clear need for a system to learn/practice sign language.

There is a wide variety of sign languages that are used by hearing-impaired individuals around the world. Each language is formed by grammar rules and a vocabulary of signs. Something that most of these languages have in common is that signs are composed by two elements: *hand postures*, i.e. the position or configuration of the fingers; and *hand gestures*, i.e. the movement of the hand as a whole. In this paper we focus on the problem of sign classification based on hand postures and hand gestures, leaving elements such as facial gestures or grammar rules for future work.

Initial work on sign language recognition has been based on sensor gloves [8] or shape descriptors for the recognition of postures. For the recognition of gestures accelerometers or colored gloves have been used to assist tracking the hand. In recent years the release of the MS Kinect device, a low-cost depth camera, has provided means to acquire relatively accurate 3D data about objects. This has been followed by a variety of prototype gestural interfaces and definitively gives an opportunity to provide an automatic solution that can alleviate the problem related to low teacher-student ratio, previously introduced. However most of these prototype gesture interfaces consider very simple gestures,

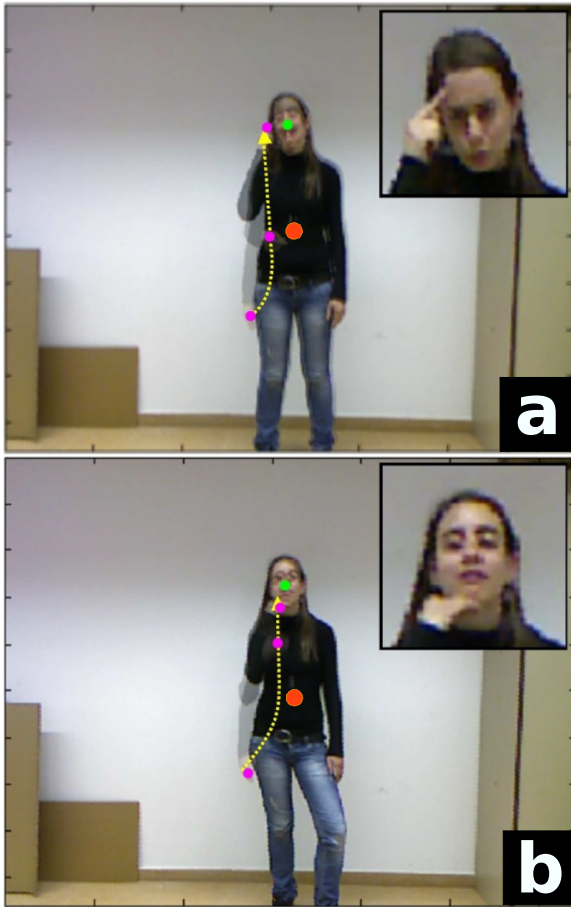


Fig. 1 Note how signs with similar global trajectories (in yellow) can be distinguished based on the relative locations of the hand (in magenta) w.r.t. the head (in green). In addition, see how the posture of the hands can help to distinguish between similar signs (see insets). Selected body part locations in color. Green: head location, magenta: right hand, orange: torso. Images taken from the ChaLearn Gestures dataset [11]. (Best viewed in color).

mostly targeted to interaction with consumer products, or employ weak gesture description methods that are not suited to accurately recognize sign language.

In this work, we consider relations between different parts of the body for the task of sign language recognition. For example, see how the global motion of the sign in Figure 1(a) is very similar to the motion of the sign in Figure 1(b). However, the relative motion of the hand (magenta) w.r.t. the head (green) is different for both signs, especially at the very end. Our method uses a first generation MS Kinect device to capture the data, in particular RGBD images to localize the different body parts. Then, each sign is represented by a combination of responses obtained from cues extracted from hand postures and hand gestures, respectively. For the problem of sign language recognition based on hand posture cues, we use shape context descriptors in combination with a multiclass Support Vector Ma-

chine (SVM) classifier to recognize the different signs. Regarding sign recognition based on cues derived from hand gestures, we use Hidden Markov Models (HMMs) to model the dynamics of each gesture. Finally, sign prediction is achieved by the late fusion of the responses of the processes for sign recognition based on hand postures and gestures, respectively. This paper extends our previous work [26] in four directions. First, we provide an extended discussion of related work, taking into account the recent literature. Second, we provide a more detailed presentation of the internals of our method, complemented by related experiments. Third, we propose and evaluate an alternative method for fusing the responses based on hand postures and gestures features, respectively. Finally, we extend our evaluation to two additional datasets.

The main contribution of this work is to show that reasoning about relations between parts of the body for the recognition of hand gestures brings improvements for hand gesture recognition and has potential for sign language recognition. This paper is organized as follows: Section 2 positions our work with respect to similar work. In Sections 3 and 4 we present the details of our method and its implementation, respectively. Section 5 presents the evaluation protocol and experimental results. In Section 6 we conclude this paper.

2 Related Work

For many years, the art of gesture recognition has been mostly focused on 2D information [1, 25, 42]. However, using this approach, there are still several challenges to be addressed, for instance, illumination change, background clutter, etc. Recently, with the advent of low-cost depth-cameras, reasoning can be focused in 3D space (e.g. [3, 23]), using jointly depth and color images. Working in the 3D space the problems of illumination change and background clutter can be reduced. In addition, the objects of interest can be isolated or segmented more accurately. Thanks to the recent development of inexpensive depth cameras, we will adopt a low-cost vision-based approach in which we use the consumer camera Kinect. Starting from this point, existing work can be divided into the four following groups:

2.1 Hands-focused Methods

Previous works in vision-based sign language recognition have mostly focused on isolating the hands and then reasoning about features extracted exclusively from them. These works formulate the sign language recognition problem either as a hand posture recognition prob-

lem or as a trajectory matching problem. For instance, for the case of hand posture-based methods, In [35], a non-rigid image alignment algorithm is proposed to enforce robustness towards hand shape variations. Furthermore, a Bayesian network formulation is used to enforce linguistic constraints between the hand shapes at start and end of a sign. In [33], Ren et al. propose a novel distance metric using RGBD images to measure the dissimilarity between hand shapes. Their method is able to distinguish slightly-different hand postures since they match the finger parts rather than the whole hand. Similarly, in [23], depth images are used to extract rotation, translation and scale invariant features which are used to train a multi-layered random forest model. This model is later used to classify a newly observed hand posture. Billiet et al. [3] present a model-based approach in which they represent the hand using pre-defined rules. Their hand model is based on a fixed number of hand components. Each component is a finger group with its associated finger pose. The hand is segmented based on depth information. Then, the RGB image is used to recognize the different hand postures. For the case of trajectory-based methods, a common practice is to track and describe the global motion of the hand, either in the 2D image space or the 3D scene space. In [25] color filtering in the HSV space is used to segment the hands in the image space. Then, the global 2D trajectories of the hands are represented as regular expressions and matched against a set of pre-defined rules representing hand gestures of interest. In [40], RGBD images collected with a first-generation MS Kinect are used to estimate the 3D location of the hands. During testing, recognition is achieved by aligning the global 3D motion trajectory of a given sign w.r.t. each sign from a pre-defined vocabulary of signs. More recently, Wang et al. [36] proposed a method where signs are described by typical posture fragments, where hand motions are relatively slow and hand shapes are stable. In addition, the 3D motion trajectory of each hand is integrated taking into account the position and size of the signer. During testing, the sequence of hand postures and the 3D trajectories are matched against a gallery of sign templates. These methods achieve good results, however they focus their reasoning on features derived from the hands in isolation. Compared to these works, our method takes into account the context (parts of the body) in which the hand trajectories occur. In addition, for the works that rely on modeling hand postures, their methods rely on an accurate construction of a hand model. However, such accurate construction may not be possible for the case of low-resolution images as is typically the case when one wants to extract gesture-based information at the same time. On the

contrary, we propose a method based on lower-level features which relaxes the requirement of high-resolution images.

2.2 Exploiting skeleton representations

Regarding hand gesture recognition, skeleton-based algorithms make use of 3D information to identify key elements, in particular the human body parts. A milestone method for the extraction of the human body skeleton is presented by Shotton et al. [34]. Ever since, the human skeleton model has been widely used for gesture recognition since this approach allows relatively accurate tracking of the joints of the body in real-time. Papadopoulou et al. [29] use the skeleton representation to compute the joint angles and angular velocities between each pair of connected parts. Then, these descriptors are used to identify action poses, such as: clapping, throwing, punching, etc. In [39], a 12-dimensional skeleton-based feature vector is defined by considering global 3D location of four joints of the skeleton (left/right elbow and left/right wrists). During testing, the label of an unknown sequence is estimated by measuring its similarity w.r.t. training sequences via Dynamic Time Warping (DTW) [28]. In [9], body pose information is encoded by computing the pairwise distances between 15 joints. In parallel, motion information is encoded by computing the Euclidean distance between pairs of joints detected at the current frame and joints detected 10 frames earlier. In addition, to encode overall dynamics of body movement, similar pairwise distances are computed between the current frames and a frame where the person is in a resting position. Finally, by using GentleBoost, the most discriminative features are identified and used for testing. Similar to these works, we also use an implementation of the algorithm from [34] to acquire the set of points of the skeleton in each frame and build a descriptor modeling the joints of the hands with respect to the joints of the other parts of the body.

2.3 Mid-level Representations for Gesture Recognition

Sign/Gesture recognition can be approached by performing classification directly from features computed on shapes (postures) or trajectories (gestures) done with the hands. However, there is a more recent trend in which these initial features are used to define mid-level representations. These representations are general enough to be used as a common vocabulary along different action/gesture classes. Furthermore, they can cope with small intra-class variations that can be introduced by

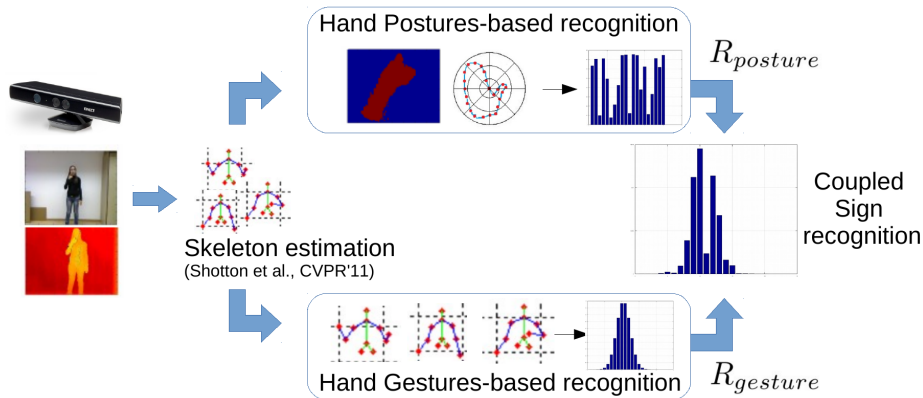


Fig. 2 Algorithm pipeline. Skeleton joints are estimated using a MS Kinect and the method from [34]. Then, the distribution over all possible sign classes is computed based on posture and gesture features, respectively. Finally, these two responses are combined and the final sign label is predicted.

different individuals performing the actions/gestures. Following this trend, Ellis et al. [9] propose a Logistic Regression learning framework that automatically finds the most discriminative canonical body pose representation of each action and then performs classification using these extracted poses. In [17], the covariance matrix between skeleton joint locations over time is used as a descriptor (Cov3DJ) for a sequence. The relationship between joint movement and time is encoded by taking into account multiple covariance matrices over sub-sequences in a hierarchical fashion. Labeled training data is encoded with these descriptors and a linear SVM classifier is trained which is later used during testing. More recently, in [32], a set of shared spatio-temporal primitives, subgestures, are detected using genetic algorithms. Then, the dynamics of the actions of interest are modeled using the detected primitives and either HMMs or DTW. Similar to the previous works we use mid-level representations to perform hand gesture classification. In this work, we first compute pairwise relations between skeleton joints for each frame in our training sequences. Then, we re-encode each of the sets of pairwise relations via K-Means, where each cluster center is a representative pose that the body can take when performing one of the gestures/actions.

2.4 Modeling Hand Gestures Dynamics

Apart from the spatial representation of hand gestures, another main problem to be solved is the temporal alignment among different sequences. Hand gestures may be understood as continuous sequences of data points or temporal series. The most modern approaches include Dynamic Time Warping (DTW), Conditional Random Fields (CRF), Hidden Markov Models (HMMs) and Rank Pooling. Despite several extensions of DTW, the disadvantage of using DTW is the heavy computational

cost involved to find the optimal time alignment path, which makes DTW practical only for small data sets. CRF is based on discriminative learning. In [6] Chung and Yang use a CRF with a threshold model to recognize the different feature vectors which are described by the angular relationship between body components in 3D space. Different from CRF, HMM is a generative method which learns how to model each class independently of the rest. In [10], Elmezain et al. propose a system to recognize the alphabet (American Sign Language) and numbers in real time by tracking the hand trajectory using HMMs. Gu et al. [15] implement a gesture recognition system using the 3D skeleton provided by the MS Kinect device. They use HMMs to model the dynamics of the training gestures, one HMM per gesture class. Very recently, Fernando et al. [13] proposed Rank Pooling, a method to perform action recognition by modeling the evolution of frame-level features over time by using ranking machines. In the context of gesture recognition, in [13] the skeleton joints are computed using the method from [34]. Then, for each frame, the relative location of each body joint w.r.t. the torso joint is computed (similar to our HD baseline, see Section 5.2). Each frame is re-encoded using the learned parameters of a ranking machine trained to order these skeleton quantized features chronologically. Finally, a SVM classifier is trained and used later during test time. In this work, we have chosen HMMs due to their remarkable performance on gesture recognition, being used in the top performing methods in previous editions of the ChaLearn Gesture Challenge [11, 16]. In the proposed method, HMMs are used to model the sequential transition between body poses acquired with Kinect during each of the signs of interest. This allows us to perform sign recognition based on cues derived from relative gestures of the hands.

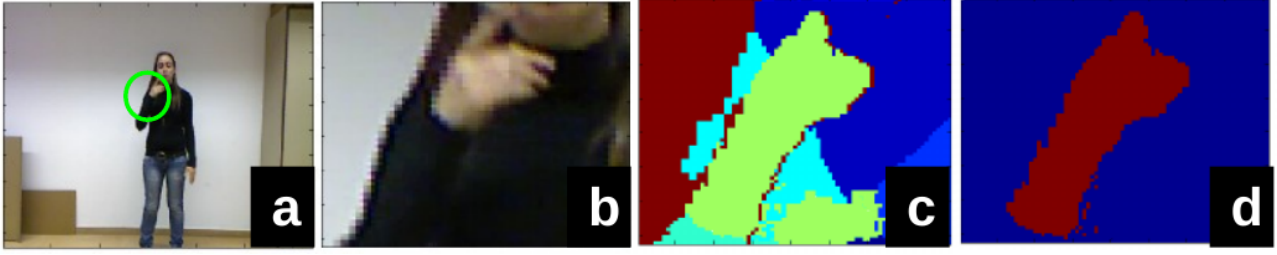


Fig. 3 Hand segmentation algorithm: (a) Original RGB image collected with kinect, (b) Cropped RGB image after spatial thresholding, (c) Projected 3D points assigned to the different parts of the body (light green:hand, cyan:arm, blue:shoulder, red:background), and (d) hand region H after applying binarization to the 2D points derived from the 3D points assigned to the parts of the hands. The largest region assigned the hand label (in red) is selected.

3 Proposed Method

The proposed method can be summarized in the following steps (see Figure 2): First, a MS Kinect device is used to capture the RGB and depth images. Based on these images we estimate the skeleton body representation using the algorithm from Shotton et al. [34]. Then, our method consists of two parallel stages: the recognition of signs based on hand posture features and the recognition of signs based on hand gesture features. Finally, the response of the recognition of signs based on hand posture features is combined with the response based on gesture features to estimate the likelihood of a given sign.

3.1 Sign Recognition based on Hand Postures

3.1.1 Hand Region Segmentation

The component based on hand posture features takes as input RGBD images and the skeleton body representation estimated using a MS Kinect device in combination with the algorithm from Shotton et al. [34]. In order to segment the hand region, the 3D world coordinate space is calculated from the depth images obtaining the (X, Y, Z) coordinates of all the points of the scene. To reduce the number of points to be processed, we perform an early spatial threshold to filter points far from the expected hand regions. To this end, all the points outside the sphere centered on the hand joint whose radius is half the distance between the joints of the hand and elbow are removed (Figure 3(a,b)). Once the amount of points has been reduced, we assign the remaining 3D points to the closest body joint, estimated via [34], using Nearest Neighbors (NN) classification (Figure 3(c)). This cluster assignment is computed in the 3D space, keeping correspondences with the pixels in the image space. Following the cluster assignment, we only keep the points that were assigned to the joints of the hands. For the case of multiple regions assigned to

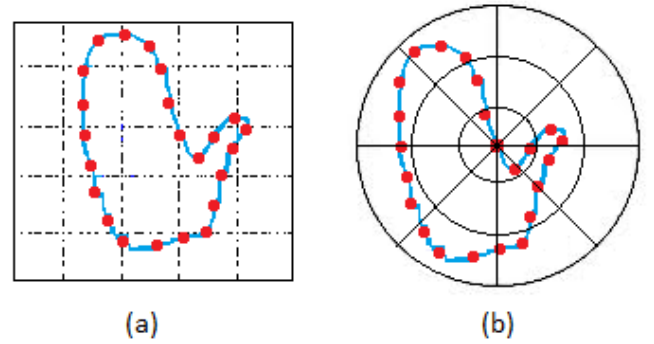


Fig. 4 Computation of Shape Context descriptors: (a) Selection of equally-spaced points on the hand region H contour, and (b) Log polar sampling (8 angular and 3 distance bins).

the hand joint, we keep the largest region. This allows our method to overcome noise introduced by low resolution images and scenarios in which the hand comes in contact with other parts of the body. In addition, we re-scale the depth images to a common 65x65 pixels patch. Finally, as Figure 3(d) shows, we binarize the re-scaled patch producing the hand regions H . Figure 3 shows the different steps starting from the input RGBD image until obtaining the hand region H .

3.1.2 Hand Posture Description

A side effect of capturing full-body images is that they result in hand regions which lack details (Figure 3(b)). Hence, a method to robustly encode information from low-resolution hand regions from images is desirable. For this reason, once we have obtained the candidate 2D regions H containing the hands, we describe the different hand postures by a Bag-of-Words representation constructed from shape context descriptors [2].

In order to compute the shape context descriptor s , we extract a number of m equally-spaced points from the contour of each binary hand region H (Figure 4(a)) obtained from the hand segmentation step. Then, using this set of points, a log-polar binning coordinate system

is centered at each of the points and a histogram accumulates the amount of contour points that fall within each bin (Figure 4(b)). This histogram is the shape context descriptor. This procedure is performed on each frame of the video sequence. Then, we define a Bag-of-Words representation p where each video is a bag containing a set of words from a dictionary obtained by vector-quantizing the shape context descriptors s via K-means. This procedure is applied for both hands of the user producing two descriptors, (p_{right}, p_{left}) , one for each hand, which are concatenated into one posture-based descriptor $p = [p_{right}, p_{left}]$.

3.1.3 Recognizing signs based on hand postures features

Once the posture descriptors p_i for all the MS Kinect video sequences have been computed, we train a multiclass SVM classifier using the pairs (p_i, c_i) composed by the concatenated posture-based descriptor p_i with its corresponding sign class c_i . We follow a one-vs-all strategy and the method from Crammer and Singer [7] to train the classifier and learn the model W .

During testing, given a video sequence captured with MS Kinect, a similar approach is followed to obtain the representation p_i based on posture features. Then, as Eq. 1 shows, the learned model W is used to compute the response $R_{posture}$ of the input video sequence over the difference sign classes, based purely on hand postures.

$$R_{posture} = W * p_i. \quad (1)$$

where p_i is the posture-based descriptor computed from the testing example and $W = [W_1, W_2, \dots, W_k]^T$ is the matrix of weights from the SVM models (one for each of the sign classes), purely based on hand posture features.

3.2 Sign Recognition based on Hand Gestures

Similarly to the hand posture component (Section 3.1), we take as input for the hand gestures component RGBD images collected with kinect and the human skeleton joints estimated using the method from [34]. The goal of this component is to infer from this skeleton a set of features that enable effective recognition of signs based on hand gestures. Towards this goal, from the initial set of 15 3D joints, we only consider a set $J = \{j_1, j_2, \dots, j_{11}\}$ of 11 3D joints covering the upper body (see Figure 5). This is due to the fact that most of the sign languages only use the upper part of the body to define their signs.

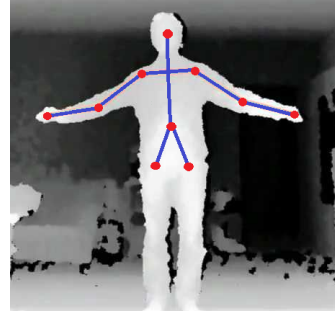


Fig. 5 Skeleton joints of the upper part of the body considered for describing signs.



Fig. 6 Relative Body Parts Descriptor (RBPD) computation. For clarity, we only show the RBPD computed for head, torso and shoulder joints for the left (green) and right (red) hands, respectively. In practice, these descriptors are computed between the 3D locations of the hands wrt. all the joints of the upper part of the body.

3.2.1 Hand Gesture Representation

Once the set of joints J have been selected, we define a descriptor to represent hand gestures based on relations between the hands and the rest of joints, or parts, of the body. This is motivated by two observations: first, because most sign languages use hands as the main, or most active, element of the signer. Second, because during different hand gestures the hands may follow similar trajectories, however these trajectories can be defined in the context of different body areas. For example, in Figure 1, even when the signs in row 1 and row 2 have a similar global trajectory (Figure 1.(a)), in yellow, the sign in row 1 involves hand contact on top of the head, while the sign in row 2 involves contact with the lower part of the head (Figure 1.(b)).

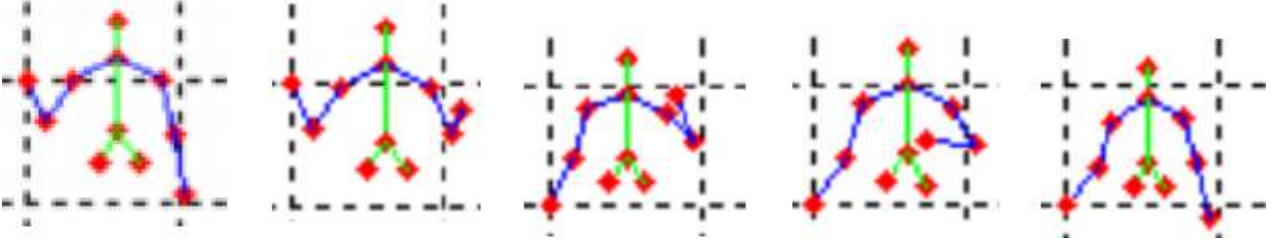


Fig. 7 Examples of cluster centers in the set of relative body poses from the whole training data of the ChaLearn dataset [11].

Given the set J of selected joints where each joint $j = (X, Y, Z)$ is defined by its 3D location. We define the Relative Body Part Descriptor (RBPDP) as $RBPDP = [\delta_1, \delta_2, \dots, \delta_m]$ where $\delta_i = (j_i - j_h)$ is the relative location of each non-hand joint j_i w.r.t. one of the hand joints j_h (Figure 6). We perform this operation for each of the two hands. The final descriptor is defined by the concatenation of the descriptors computed from each hand $RBPDP = [RBPDP_{right}, RBPDP_{left}]$. Notice that the length of this descriptor is 66 since we are considering 11 parts of the body including the hands. Furthermore, note that, the user can be at different locations with respect to the visual field of the camera and consequently there might be considerable variation in X, Y and Z coordinates. However, by building the proposed descriptor, considering relative locations between the hands w.r.t. body, we achieve some level of invariance towards translation in the location of the user. Finally, until now, the estimated input descriptor $RBPDP$ constitutes the observation at a specific frame. In order to extend this frame-level representation to the full gesture sequence we compute this descriptor for each of the n frames of the video $g = [RBPDP_1, RBPDP_2, \dots, RBPDP_n]$.

3.2.2 Mid-level Feature Encoding

Up to this point, every video sequence is represented as a sequence of displacement vectors between body parts ($RBPDP$). Each vector being computed independently of the user and the sign class. However, even when focusing on single sign class, different users may introduce small variations to the sign they perform. Likewise, some sign classes may share some characteristics at the gesture level. To address these issues, we re-encode the $RBPDP$ s by using a mid-level representation that can be shared between both users and sign classes. To this end, we compute the $RBPDP$ s from all the frames of the training sequences, z-normalize them, and cluster them using K-means with. This K value was obtained from running the pipeline in the validation set. Then, each video is re-encoded by the sequence of cluster centers w_i that its corresponding $RBPDP$ s are assigned to. As

a result, each gesture will now be represented by a sequence of centers w . These cluster centers w are stored for later use during the testing stage. See Figure 7 for some examples of the cluster centers w .

3.2.3 Recognizing signs based on hand gesture features

In this paper, we model the dynamics of the hand gestures using left-right Hidden Markov models (HMMs). Specifically, we train one HMM per sign class. HMMs are a type of statistical model which are characterized by the number of states in the model, the number of distinct observation symbols per state, the state transition probability distribution, the observation symbol probability distribution and the initial state distribution. In our system, the training observations (o_1, o_2, \dots, o_n) are the hand gestures represented as a sequence of centers estimated from the encoding step. These observations o_i are collected per sign class c_i and used to train each HMM. The state transition probability of each model is initialized with the value 0.5 to allow each state to begin or stay on itself with the same probability. The number of states is different for each model and was determined using validation data. In addition, the number of distinct observation symbols of the models is equal to the number of centers K . Furthermore, in order to ensure that the models begin from their respective first state, the initial state distribution gives all the weight to the first state. Finally, the observation symbol probability distribution matrix of each model is uniformly initialized with the value $1/K$, where K is the number of distinct observation symbols. During training, for each model, the state transition probability distribution, the observation symbol probability distribution and the initial state distribution are re-adjusted by using the Baum-Welch algorithm [18]. Once the different HMMs have been trained for each sign class c_i , the system is then ready for sign classification. During testing, given a gesture observation g , sequence of encoded centers, and a set of pre-trained HMMs Ω , our method selects the class of the model Ω_k that maximizes the likelihood $p(c|g)$ of class c based on gestures features (Eq.2). In this paper we refer to such likelihood $p(c|g)$

as the sign response $R_{gesture}$ based, purely, on hand gesture features.

$$c = \arg_k \max p(k|g) = \arg_k \max(\Omega_k(g)) \quad (2)$$

3.3 Coupled Sign Language Recognition

For each RGBD sequence captured with MS Kinect, the earlier components of the system compute the $R_{posture}$ and $R_{gesture}$ responses over the sign classes based on posture and gesture features, respectively. In order to obtain a final prediction, we define the coupled response R by late fusion of the responses $R_{posture}$ and $R_{gesture}$. To this end, given a set of validation sequences, for each example sequence we compute the responses based on the postures $R_{posture}$ and gestures $R_{gesture}$. In addition, for each example, we define the coupled descriptor $R = [R_{posture}, R_{gesture}]$ as the concatenation of the two responses. Then, using the coupled descriptors - class label pairs (R_i, c_i) from each validation example we train a multiclass SVM classifier using linear kernels. This effectively learns the optimal linear combination of $R_{posture}$ and $R_{gesture}$. During testing, the sign class \hat{c}_i is obtained as:

$$\hat{c}_i = \arg_{c_k} \max(\omega_k \cdot R_i). \quad (3)$$

where R_i is the coupled response computed from the testing data and $\omega = [\omega_1, \omega_2, \dots, \omega_k]^T$ are the weight vectors from the SVM models.

In addition to the previous SVM-based method to perform a linear combination of the responses, we explore the performance of an alternative probabilistic method [30] to combine the responses. Given the coupled descriptors - class label pairs (R_i, c_i) from each validation example, the sign class \hat{c}_i or R_i is the MAP estimate by applying the Bayes rule:

$$\hat{c}_i = \arg_{c_k} \max p(R_i|c_k)p(c_k), \quad (4)$$

where the class likelihoods $p(R_i|c_k)$ are computed using Kernel Density Estimation (KDE) and the priors $p(c_k)$ are obtained from the occurrence of sign class c_k on the validation data.

4 Implementation Details

In this section we provide some implementation details in order to ease the reproducibility of the method proposed in this paper.

As mentioned in Section 3.1.2 we define our posture-based representation from shape context descriptors [2]. In our experiments, we use a pseudo log-polar sampling mask with 12 angular and 5 distance bins (with an inner radius 6 pixels and an outer radius 32 pixels) delivering a 60 dimensional histogram for each of the sampled points. We combine the inner part of the log polar mask used to build the shape context descriptor into one bin since there is some evidence [27] that combining this inner part produces improved results. For this reason, the length of our shape context descriptor is reduced to 49 dimensions. Then, each shape context descriptor is normalized dividing each element of the descriptor by the sum all the elements of the descriptor. Once the hand region H has been segmented, the shape descriptor is computed on a total of 20 equally spaced points. When performing K-means a value of $K=100$ was used since that value gave the best performance in the validation set. In addition, during SVM training (Section 3.1.3), at the posture stage, we use 3-fold cross validation and a cost value $C = 0.8352$.

During the coupled sign recognition stage, Section 3.3, we train the SVM models via 3-fold cross validation with a cost value $C = 0.7641$. In our implementation, we use the Liblinear [12] for SVM training and classification. We perform multiclass classification following a one-vs-all strategy and the method from Crammer and Singer [7] to train the models. For the case of the alternative probabilistic response-fusion method based on KDE, we use the Online Kernel Density Estimation (oKDE) variant proposed in [21, 22]. However, since no online learning/estimation is required, we apply low compression and construct the initial estimator from the whole set of training examples. In consequence, we only keep its variable multivariate properties for kernel density estimation.

5 Evaluation

In this section, we present the experimental protocol followed to evaluate the performance of the proposed method. We divide our evaluation into five subsections aimed at analyzing different aspects of the proposed method. To this end, we evaluate its performance when only considering posture-based features (Section 5.1), its performance when only considering gestures-based features (Section 5.2), the combination of both posture and gesture features (Section 5.3), a comparison of the proposed method w.r.t. state-of-the-art methods (Section 5.4), and its computation time (Section 5.5).

We evaluate our approach on the ChaLearn Multimodal Gesture Recognition Challenge 2013. This dataset was introduced in [11]. It contains 20 Italian cultural /

anthropological signs produced by a total of 27 subjects. It provides RGBD images captured with a MS Kinect device plus the skeleton joints estimated using the method from [34]. All the examples in the training and validation sets are annotated at frame level indicating the beginning and ending frame of each sign. Since the focus of our work is towards sign classification rather than sign detection, we organize our data in isolated sign sequences. For the sake of comparison with recent work [31, 39, 41], we use the original training set of the dataset for training and the original validation set of the dataset for testing. This is split in such a way that ensures that a subject whose data occurs in the training set, does not occur in the testing set. In addition, we split the training set into two subsets, one subset for training and one subset for validation purposes. Moreover, different from the original, Levenshtein distance, performance metric used in the challenge [11], we report results using as performance metric mean precision, recall and F-Score. In addition, for reference, we present results on the original testing set of the ChaLearn Gesture dataset which annotations were kindly provided by its organizers.

Additionally, we also conduct experiments on the MSR Action3D dataset [24]. It includes 20 classes of actions. Each action was performed by 10 subjects for three times. This dataset was captured at 15 fps with a resolution of 320x240. It is composed by 23797 frames of RGBD images for 402 action sequences. For the sake of comparison we follow a similar evaluation protocol as proposed in [24, 38] to split the data into training and testing sets. We report performance in terms of mean accuracy, precision, recall and F-Score. Different from the ChaLearn Gestures dataset, this dataset is more oriented towards general actions, e.g. “pick up and throw”, “golf swing”, “hand clap”, “hammer”, etc. However, we will only focus only on the joints of the upper part of the body for the description of hand gestures.

Finally, we perform experiments on the MSRC-12 dataset [14]. This dataset is captured at 30 fps and composed of 594 sequences (719359 frames) from 30 subjects performing 12 gestures. We conduct experiments on the MSRC-12 dataset, following the protocol from [9, 17]. Different from the ChaLearn dataset, the MSRC-12 dataset does not include RGBD images for the sake of anonymity. For this reason, we cannot report results for the combined method on MSRC-12, since RGBD images are required for the processing of hand postures.

5.1 Sign Recognition based on Hand Postures

In this experiment we evaluate the performance of the method at recognizing signs based purely on features derived from hand postures. Table 1 presents the mean performance of our method when only considering postures computed from hand postures (Section 3.1). Figure 10 (first column) shows the confusion matrix of this experiment in the test set.

Table 1 Hand Posture-based recognition mean performance.

ChaLearn (val.) dataset [11]		
Precision	Recall	F-Score
0.42	0.35	0.38

ChaLearn (test.) dataset [11]		
Precision	Recall	F-Score
0.34	0.33	0.34

MSR Action3D dataset [24]		
Precision	Recall	F-Score
0.40	0.40	0.40

Discussion: recognition based on hand posture features has an average F-Score of 0.38, 0.34 and 0.40, on the ChaLearn (validation and testing sets) and MSR Action3D datasets, respectively. This low average is due to the fact that: (1) the signs were captured at a distance around two to three meters from the camera obtaining images with poor resolution, specially for the regions that cover the hands. In addition, it should be noted that the hand, compared with the complete human body, is a smaller deformable object and more easily affected by segmentation error. (2) On many of the signs, the hands come into contact or get very close to the body (see Figure 1(b)) making it difficult to obtain a good segmentation and introducing error in the features computed from the hand region. (3) Some of the signs are defined with very similar sequences of hand postures, being only different in one or two hand postures along the sequence (in this case, the most significant hand posture(s) that define the sign) easily leading to sign miss classification. If we compare our shape context-based method with other methods for hand posture recognition [3, 20, 33], we notice that our method is better suited for these low-resolution images. This is due to the fact that our method does not rely on the construction of a more detailed hand model which is a difficult task on low-resolution images like the ones of the ChaLearn gestures dataset. On the contrary, our method is able to leverage posture features from low-resolution images removing the requirement of a detailed hand model.

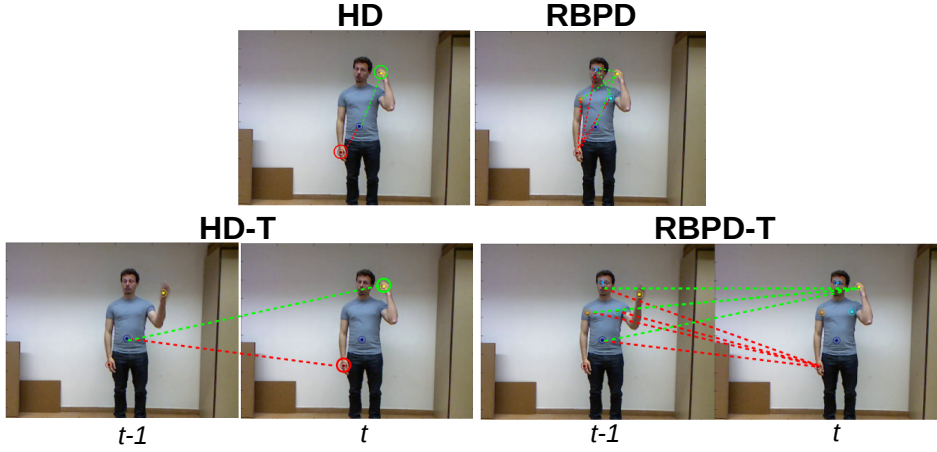


Fig. 8 Evaluated methods to model hand gestures. The purely spatial descriptors which operate at the frame level: hand descriptor (*HD*) and Relative Body Parts Descriptors (*RBPd*), and their time-extension counterparts, *HD-T* and *RBPd-T*, which operate between frames at different time stamps.

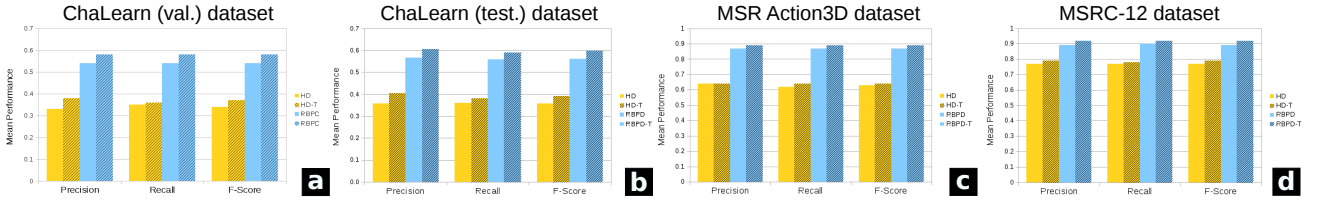


Fig. 9 Gesture-based recognition mean performance on the validation (a) and test (b) set of the ChaLearn gestures dataset [11], the MSR Action3D dataset (c) [24], and the MSRC-12 dataset (d) [14]. Note how the performance of purely focusing on global hand trajectories (*HD*, *HD-T*), presented in yellow color, is much lower than the performance of our method considering body part relations (*RBPd*, *RBPd-T*), presented in light blue color. Furthermore, note how considering the time extension of the descriptors (*HD-T*, *RBPd-T*) brings a small improvement over their purely spatial counterparts (*HD*, *RBPd*).

5.2 Sign Recognition based on Hand Gestures

In this experiment we focus on the recognition of signs based on hand gestures. We evaluate four methods (see Figure 8) to model the gestures: a) the *RBPd* descriptor proposed in Section 3.2; b) the *RBPd-T* descriptor which is similar to *RBPd*, however, in this descriptor the relations between the hands and the other parts of the body are estimated taking into account the hand locations in the current frame and the location of the other parts in the next frame. As a result, this descriptor not only takes into account spatial relations but implicitly adds temporal features; c) the *HD* descriptor which only considers the location of the hands w.r.t. the torso location; and d) *HD-T*, a time extension of *HD*. The last two methods, *HD* and *HD-T*, are based on hand trajectories since we only follow the location of the hands over time. Similar to *RBPd*, we train HMMs (Section 3.2.3) using these methods, *RBPd-T*, *HD*, and *HD-T*, for gesture representation. From these methods, we take the top performing *RBPd-T* for further experiments. Figure 9 and Table 2 show the mean performance of each of these methods to model gestures in the evaluated datasets. Figure 10(second column) shows the confusion matrix of recognizing signs based on hand gesture features.

Table 2 Gesture-based recognition mean performance.

	ChaLearn (val.) dataset [11]		
	Precision	Recall	F-Score
HD	0.33	0.35	0.34
HD-T	0.38	0.36	0.37
RBPd	0.54	0.54	0.54
RBPd-T	0.58	0.58	0.58

	ChaLearn (test) dataset [11]		
	Precision	Recall	F-Score
HD	0.36	0.36	0.36
HD-T	0.40	0.38	0.39
RBPd	0.57	0.56	0.56
RBPd-T	0.61	0.59	0.60

	MSR Action3D dataset [24]		
	Precision	Recall	F-Score
HD	0.64	0.62	0.63
HD-T	0.64	0.64	0.64
RBPd	0.87	0.86	0.87
RBPd-T	0.89	0.89	0.89

	MSRC-12 dataset [14]		
	Precision	Recall	F-Score
HD	0.77	0.77	0.77
HD-T	0.79	0.78	0.79
RBPd	0.89	0.90	0.89
RBPd-T	0.92	0.92	0.92

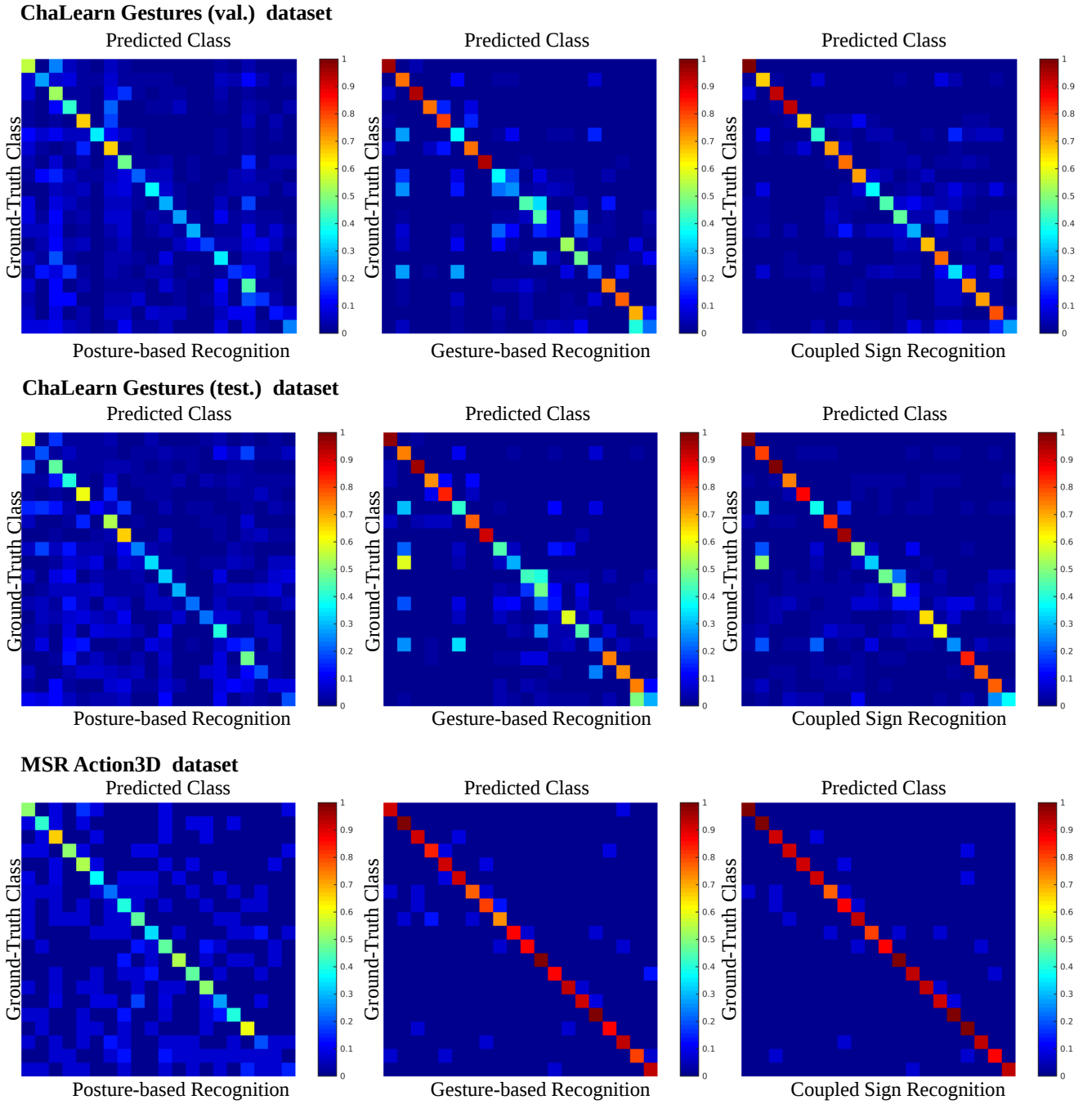


Fig. 10 Confusion matrices for sign recognition based on responses computed from Hand Postures (first column), Hand Gestures (second column) ($RBPD-T$), and late fusion (probabilistic) of hand postures and gestures responses (third column).

Discussion: A quick inspection to Figure 9 reveals that taking into account relations between different body parts when modeling hand gestures brings improvements over methods that only consider global hand trajectories for sign recognition. This is supported by an improvement of 24 percentage points (pp) higher mean F-Score of $RBPD$ over HD on the ChaLearn Gestures and MSRC-12 datasets, Note that these datasets are more sign language oriented. For the case of the MSR

Action3D dataset, this improvement of performance is around 24 pp, still confirming that this collective reasoning about parts of the body brings improvement to action recognition. Compared to this, the differences between the performance of the $HD, RBPD$ and $HD-T, RBPD-T$, respectively, seem to be minimal. The time extensions $HD-T$ and $RBPD-T$ seem to bring higher improvement the closer the problem is to a sign language recognition setting. For example, in the ChaLearn

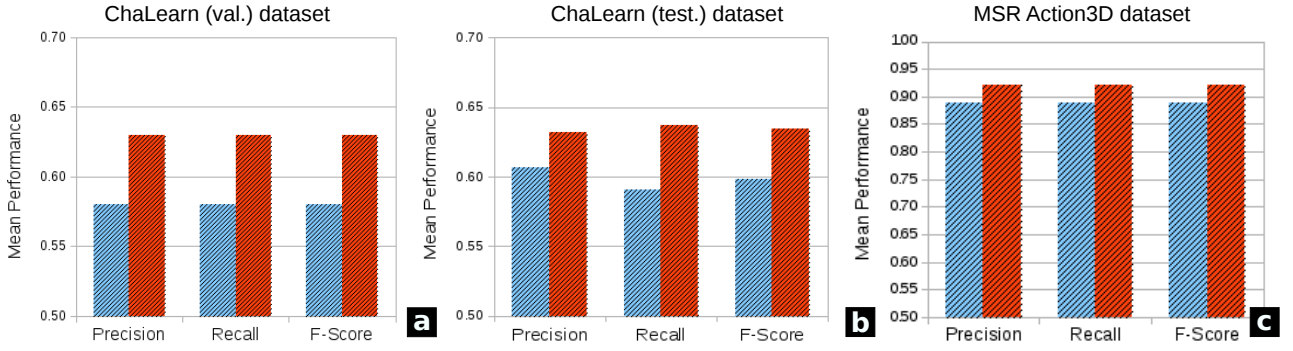


Fig. 11 Mean F-Score when performing sign recognition when considering only gesture-based features (light blue) and when considering both hand postures and hand gestures features (orange) on the ChaLearn [11] (a,b) and MSR Action3D [24] (c) datasets.

dataset, it brings an improvement of ~ 4 pp while in the MSRC-12 dataset this improvements drops to ~ 2 pp. This further drops to ~ 1 for the more general action classes of the the MSR Action3D dataset (see Table 2). As Figure 10(second column) shows the *RBPD-T* is able to recognize some signs with very high accuracy, e.g. signs 1, 3 and 8 of the ChaLearn dataset. This is due to the fact that these signs are more different from the other signs such that their respective hand gesture representations, sequence of cluster centers, are more unique. As Figure 10(second column) also shows the main problems are the confusions of the signs 6, 9, 10 and 16 with 2 and signs 11, 13 and 15 with sign 12 due to the similarities between the gestures of those signs being differentiated mostly by particular hand postures (see Figure 12 for a qualitative example). For the case of the MSR Action3D dataset Figure 10(second column, last row) , where hand postures take a secondary role, we can notice that hand gestures-based features alone can produce very good performance.

5.3 Coupled Sign Recognition

In this experiment we evaluate the performance of the coupled response $R = [R_{posture}, R_{gesture}]$ based on hand posture and hand gesture features as described in Section 3.3. We compare the performance provided by the two methods presented in Section 3.3 to perform the combination of the responses based on hand postures and hand gestures features, respectively. Table 3 presents the performance of different response combination methods on the ChaLearn gestures dataset and the MSR Action3D dataset. As mentioned earlier, no performance on Coupled Sign Recognition is presented for the MSRC-12 dataset since no hand posture information can be extracted from it. Figure 10(third column) shows the confusion matrix for the combination of the responses in the evaluated datasets.



Fig. 12 Some of the confusing signs from the ChaLearn gestures dataset [11] when only considering gesture-based information. Notice how the motion of the hand is very similar along the different signs. However, they can still be differentiated by the posture of the hand (marked by the red box).

Table 3 Coupled recognition mean performance. Gestures features are based on *RBPD-T*.

Fusion Method	ChaLearn (val.) dataset [11]		
	Precision	Recall	F-Score
Linear Combination	0.61	0.62	0.62
Probabilistic	0.63	0.63	0.63

Fusion Method	ChaLearn (test.) dataset [11]		
	Precision	Recall	F-Score
Linear Combination	0.63	0.62	0.62
Probabilistic	0.63	0.64	0.63

Fusion Method	MSR Action3D dataset [24]		
	Precision	Recall	F-Score
Linear Combination	0.91	0.91	0.91
Probabilistic	0.92	0.92	0.92

Table 4 Comparison with the State of the Art in chronological order. Mean performance over all the 20 sign classes of the ChaLearn 2013 dataset [11].

	Precision	Recall	F-Score
Wu et al., [39]	0.60	0.59	0.60
Yao et al., [41]	-	-	0.56
Pfister et al., [31]	0.61	0.62	0.62
Fernando et al., [13]	0.75	0.75	0.75
Ours (linear comb.)	0.61	0.62	0.62
Ours (probabilistic comb.)	0.63	0.63	0.63

Discussion: At first sight, as Figure 11 shows, the combination of responses, based on hand postures and gestures features, outperforms the overall performance of the method when considering only hand gestures. In addition, it can be noted from the confusion matrices (Figure 10) of both methods that confusion between sign classes is reduced showing the complementarity of both responses, based on postures and gestures, respectively. This is to be expected since some ambiguous cases can be clarified by looking at the relations between parts of the body (see row 1 vs row 2 of Figure 1(a)). Likewise, other ambiguous cases can be clarified by giving more attention to the hand postures (Figure 12 third column). In addition, we can notice in Table 3 that the proposed methods to combine the responses based on hand postures and hand gestures have a similar performance. Nevertheless, the probabilistic method based on KDE provides an improvement ~ 1 pp over the method based on linear combination of the responses.

5.4 Comparison w.r.t the state-of-the-art

Given the observations made in the previous experiments, in this experiment we select the top-performing method, i.e. *RBPD-T* for gesture modeling and probabilistic combination of responses, and used it for comparison w.r.t. the state-of-the-art. For the case of the ChaLearn gestures (val.) dataset we compare with recent work [13, 31, 26, 39, 41]. We report results in Table 4 using as performance metric mean precision, recall and F-Score. Similarly, for the MSRC-12 dataset, we compare with [9, 17]. For the case of the MSR Action3D dataset, we follow the evaluation protocol from [24, 19, 37, 38] and compare the performance of our method with the one reported by those methods, respectively. Table 5 reports the results in terms of Mean Accuracy.

Discussion: Compared to [39], the method that was ranked 1st in the Multi-modal Gesture Recognition Challenge in 2013 [11] (when only using image/video data), our combined method achieves an improvement of ~ 4 F-Score pp over their method (Table 4). Furthermore

Table 5 Comparison with the State of the Art in chronological order. Mean Accuracy over all the 20 classes of the MSR Action3D 2013 dataset [24].

	Accuracy
Li et al., [24]	0.747
Wang et al., [38]	0.882
Wang et al., [37]	0.862
Ellis et al., [9]	0.657
Hussein et al., [17]	0.905
Jetley et al., [19]	0.838
Ponce-López et al., [32]	0.950
Ours (linear comb.)	0.908
Ours (probabilistic comb.)	0.919

Table 6 Comparison with the State of the Art in chronological order. Mean Accuracy over all the 12 gesture classes of the MSRC-12 dataset [14].

	Accuracy
Ellis et al., [9]	0.887
Hussein et al., [17]	0.903
Ours (RBPD-T)	0.919

our method is still superior by ~ 7 pp over the F-Score performance reported by the recent method from [41]. This is to be expected since our method explicitly exploits information about hand postures, which [41] ignores. This last feature makes the proposed method more suitable to address sign language recognition where hand posture information is of interest. Even more, our method has a comparable performance (1 pp improvement on performance) to the method from [31], which also considers hand posture information. However, different from [31], our method does not rely on face detection and skin segmentation in order to localize the hand regions. Compared to the just-published method from [13], our method achieves inferior performance (~ 13 pp lower F-Score). The method from [31] uses hand trajectories and the method from [4, 5] for hand posture modeling. This is closer to the *HD* method that we evaluated which, in our experiments, produced suboptimal performance. In the fully supervised case, [31] achieves comparable performance as our method. This suggests that the method from [4, 5] (for hand posture modeling) is superior to the one used in our work. We will consider combining our relations-based method, for gesture-based recognition, with the method from [4, 5], for posture-based recognition, as future work. We expect this will improve the precision of the posture module which affects the combination of responses; especially in cases where signs have similar gestures but slightly different postures. In addition, different from our method, [13] considers neither relations between parts of the body nor hand posture information. In [13], hand joints are normalized w.r.t. the torso location. This shows that using ranking machines is in-

deed a powerful mechanism for modeling the dynamics of the gestures. These observations show a strong potential on the combination of advanced methods for hand posture modeling [5], powerful mechanisms to model the dynamic of hand gestures [13] and the more detailed relational descriptions proposed in this work, for the task of sign language recognition.

For the case of the MSR Action3D dataset, our method has 3 pp superior accuracy compared to the method from [38] which uses a linear combination of mined actionlets which are conjunction of the features from a subset of the joints of the body. Our method based on the linear combination of the postures and gestures responses is comparable to the method from [17] where they use a more expensive covariance descriptor to relate the body joints. However, our method based on probabilistic combination of the responses produces an improvement of ~ 2 over [17]. Even though our method is designed for sign language recognition, it has a comparable performance (3 pp lower) to the method recently proposed in [32] on the task of general action recognition in this dataset. Compared to the method from [32], our method does not require multiple evolution of its models. However, given that [32] achieves a performance of 0.95 accuracy from a baseline of 0.71, it would be interesting to investigate the performance that can be achieved by our method (baseline accuracy: 0.92) when integrating such evolutionary steps to its models.

On MSRC-12, our method achieves 3 pp over the accuracy reported in [9] which is focused on a feature vector of pairwise joint distances between frames. Furthermore, in this dataset we observe a similar trend as in the MSR Action3D dataset where our method is slightly superior to the method proposed in [17].

5.5 Computation Time

In order to verify the potential of the proposed method for interactive applications, we computed the average processing time during inference. Our experiments were performed on a single core 2.2 GHZ CPU computer with 8 GB of RAM using un-optimized Matlab code. We summarize the processing times, in seconds, of the different stages of our method in Table 7 and Figure 13.

As can be seen in Table 7, if done sequentially, inference has an average runtime of 0.26415 seconds from which 0.21993 seconds are spent on the computation of the posture descriptor. Since the focus of this work is on the gesture part, the posture module can be improved in future work by faster, and more effective, methods for hand posture modeling.

Table 7 Average and accumulated processing times (in seconds) for each of the different stages of the proposed method. Notice how the stage related to hand postures takes 0.22011 seconds of total time (0.26415 seconds).

Stage	Process	Proc. time	Accum. time
Postures	Descr. Comp.	0.21993	0.21993
	Classification	0.00018	0.22011
Gestures	Descr. Comp.	0.02417	0.24428
	Classification	0.01726	0.26154
Combination	Descr. Comp.	0.00213	0.26367
	Classification	0.00048	0.26415
Total time			0.26415

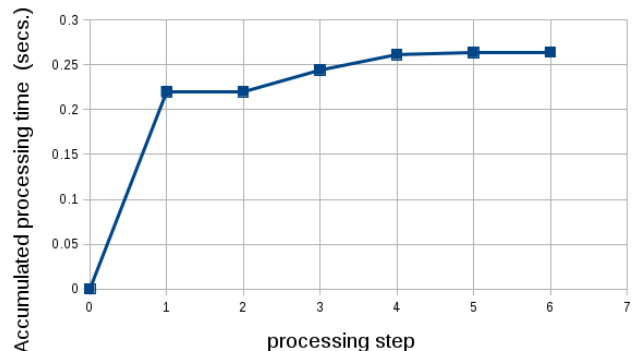


Fig. 13 Average Processing times.

6 Conclusion and Future Work

We presented a method mainly targeted for sign language recognition. The proposed method focuses on representing each sign by the combination of responses derived from hand postures and hand gestures. Our experiments proved that modeling hand gestures by considering spatio-temporal relations between different parts of the body brings improvements over only considering the global trajectories of the hands. In addition, the proposed method introduces a descriptor for hand postures that is flexible to operate on low-resolution images and that will take advantage of high-resolution images.

Future work will focus on three aspects: First, consider state-of-the-art methods to model action dynamics to describe the dynamics of hand postures and hand gestures for each sign class. Second, shift the focus of this work towards sign localization/detection. Third, consider additional features of sign languages such as grammars and facial-related gestures. Taking into consideration these other characteristics will permit the proposed system to develop into a more realistic sign language recognition system.

References

1. A. A. Argyros and M. I.A. Lourakis. *Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera*. ECCV, 2004.
2. S. Belongie and J. Malik. *Matching with Shape Contexts*. IContent-based Access of Image and Video Libraries. Proceedings. IEEE Workshop on, pages 20–26, 2000.
3. L. Billiet, J. Oramas, M. Hoffmann, W. Meert, and L. Antanas. *Rule-based hand posture recognition using qualitative finger configurations acquired with the Kinect*. ICPRAM, 2013.
4. P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Upper body detection and tracking in extended signing sequences. *IJCV*, 95(2):180–197, 2011.
5. P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *CVPR*, 2009.
6. H. Y. Chung and Hee-Deok. *Conditional random field-based gesture recognition with depth information*. Optical Engineering, Volume 52, 2013.
7. K. Crammer and Y. Singer. On the algorithmic implementation of multi-class svms. In *JMLR*, volume 2, pages 265–292, 2001.
8. L. Dipietro, A. M. Sabatini, and P. Dario. A survey of glove-based systems and their applications. *Trans. Sys. Man Cyber Part C*, 38(4):461–482, 2008.
9. C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola, Jr., and R. Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. In *IJCV*, volume 101, pages 420–436, 2013.
10. M. Elmezain, A. Al-Hamadi, and B. Michaelis. *Hand Gesture Recognition Based on Combined Features Extraction*. World Academy of Science, Engineering and Technology. Vol.3, 2009.
11. S. Escalera, J. Gonzalez, X. Baro, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. J. Escalante. Multi-modal gesture recognition challenge 2013: dataset and results. In *ICMI Workshops*, 2013.
12. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. *LIBLINEAR: A Library for Large Linear Classification*, volume 9. 2008.
13. B. Fernando, E. Gavves, J. Oramas M., A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.
14. S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In *CHI*, 2012.
15. Y. Gu, H. Do, Y. Ou, and W. Sheng. *Human Gesture Recognition through a Kinect Sensor*. ICRB, 2012.
16. I. Guyon, V. Athitsos, P. Jangyodsuk, H. Escalante, and B. Hamner. Results and analysis of the chlearn gesture challenge 2012. In *ICPR*, 2012.
17. M. E. Hussein, M. Torki, M. A. Gawayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, 2013.
18. F. Jelinek, L. Bahl, and R. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21(3):250–256, 1975.
19. S. Jetley and F. Cuzzolin. 3d activity recognition using motion history and binary shape templates. In *ACCV 2014 Workshops*, pages 129–144, 2014.
20. C. Keskin, F. Krac, Y. E. Kara, and L. Akarun. *Real Time Hand Pose Estimation using Depth Sensors*. ICCV Workshops, 2011.
21. M. Kristan and A. Leonardis. Online discriminative kernel density estimator with gaussian kernels. *IEEE T. Cybernetics*, 44(3):355–365, 2014.
22. M. Kristan, A. Leonardis, and D. Skocaj. Multivariate online kernel density estimation with gaussian kernels. *Pattern Recognition*, 44(10-11):2630–2642, 2011.
23. A. Kuznetsova, L. Leal-Taix, and B. Rosenhahn. Real-time sign language recognition using a consumer depth camera. 2013.
24. W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Human Communicative Behavior Analysis Workshop at CVPR*, 2010.
25. J. Oramas M., A. Moreno, and K. Chiliza. Potential benefits in the learning process of ecuadorian sign language using a sign recognition system. *e-Minds*, 2(7), 2011.
26. M. Martínez-Camarena, J. Oramas M, and T. Tuytelaars. Towards sign language recognition based on body parts relations. In *ICIP*, 2015.
27. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005.
28. M. Muller. Dynamic time warping. information. In *Retrieval for Music and Motion*.
29. G. Papadopoulos, A. Axenopoulos, and P. Daras. Real-time skeleton-tracking-based human action recognition using kinect data. In *Multimedia Modeling*, 2013.
30. R. Perko and A. Leonardis. A framework for visual-context-aware object detection in still images. *Computer Vision and Image Understanding*, 114(6):700–711, 2010.
31. T. Pfister, J. Charles, and A. Zisserman. Domain-adaptive discriminative one-shot learning of gestures. In *ECCV*, 2014.
32. V. Ponce-López, H. Escalante, S. Escalera, and X. Baró. Gesture and action recognition by evolved dynamic sub-gestures. In *BMVC*, 2015.
33. Z. Ren, J. Yuan, J. Meng, and Z. Zhang. *Robust Part-Based Hand Gesture Recognition Using Kinect Sensor*. TMM, 2013.
34. J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. *Real-time human pose recognition in parts from single depth Images*. CVPR, 2011.
35. A. Thangali, J. P. Nash, S. Sclaroff, and C. Neidle. Exploiting phonological constraints for handshape inference in asl video. In *CVPR*, 2011.
36. H. Wang, X. Chai, and X. Chen. Sparse observation (so) alignment for sign language recognition. *Neurocomputing*, 175, Part A:674 – 685, 2016.
37. J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *ECCV*, pages 872–885, 2012.
38. J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297, 2012.
39. J. Wu, J. Cheng, C. Zhao, and H. Lu. Fusing multi-modal features for gesture recognition. In *ICMI*, 2013.
40. Y. Lin Z. Xu Y. Tang X. Chen X. Chai, G. Li and M. Zhou. Sign language recognition and translation with kinect. In *FG*, 2013.
41. A. Yao, L. Van Gool, and P. Kohli. Gesture recognition portfolios for personalization. In *CVPR*, 2014.
42. L. Yun and Z. Peng. *An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs*. WCSE, 2009.